

SAGA-LS 制御系の現状と更新

岩崎 能尊、高林 雄一、金安 達夫、江田 茂

九州シンクロトロン光研究センター

九州シンクロトロン光研究センター加速器の制御システムは、加速器運用当初より Windows PC および National Instruments LabVIEW を用いて順次構築されてきた。リニアック構成機器類をデジタル制御とした現在、半固定的なマニュアルゲートバルブ、常時電源 ON で印加電圧が固定のイオンクリアリング電源といった一部の機器を除き、ほぼ全ての機器類をデジタル制御あるいはモニタするに至った。各機器類を制御するサーバ PC と機器類間とのリモート I/O には必要に応じ Yokogawa PLC あるいは National Instruments FieldPoint を用い、またサーバ PC と機器類間の通信インターフェースとしては Ethernet、 GPIB、 RS232C、（一部計測機器用に USB）を用いている。サーバ PC とクライアント PC 間の通信には ActiveXCA[1]を用いることで、基本的に任意のクライアント PC からの EPICS Process Variable を用いたアクセスが可能である（超伝導ウィグラー用電磁石電源のように複数 PC によるマルチアクセスや他の機器との連携操作の必要性のない一部の機器類に対してはシングル PC 制御に留めている）。また、データベースには MySQL を使用し、機器類の設定値あるいは真空度といったモニタ値の記録を 30 秒ないし 1 分間隔でログしている。SAGA-LS 加速器制御システムは基本的には機能分散型のクライアント&サーバモデルに基づくものであるが、近年、一部の加速器構成機器類制御プログラムをシーケンシャルに接続することにより、1 台の PC で複数の機器を連動制御させるプログラムの運用も試みている。このプログラムからは加速器インターロックシステムの操作も可能である。このような集約的なプログラムは通信障害が発生した場合などには特に有効な制御方法である。制御に用いている Windows OS は、PC の購入時期により、Windows 2000、XP、7 と上位バージョンのものに順次更新してきた。バージョンの異なる OS が混在した環境であるが、現在までに OS の違いによる制御上の障害は生まれていない。しかし、稼動させているサーバプログラムの種類にもよるが、数ヶ月に一度程度システムをリブートする必要があることがある。この問題によりユーザ運転を中断した例は過去にないが、長期的な制御システムの安定性確保はひとつの課題である。Windows OS 環境下における制御システムの安定化のためには PC リブートのルーチン化が簡易で有効ではあるが、原因解明のためメモリーリークの状況などの観測も続けている。

参考文献

[1] ActiveX CA Tools,

<http://www.slac.stanford.edu/grp/cd/soft/epics/extensions/ActiveXCA/readme.htm>

（メモ）

SAGA-LS制御系の現状と更新

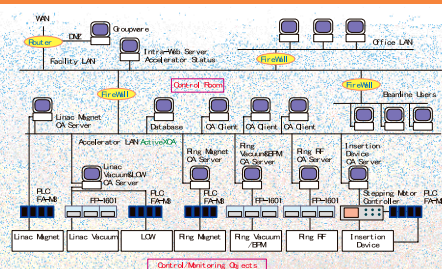
岩崎 能尊、高林 雄一、金安 達夫、江田 茂

九州シンクロtron光研究センター

九州シンクロtron光研究センター加速器の制御システムは、加速器運用当初よりWindows PCおよびNational Instruments LabVIEWを用いて順次構築されてきた。リニアック構成機器類をデジタル制御とした現在、半導体的なマニュアルゲートバルブ、常時電源ONで印加電圧が固定のイオンクリアリリング電源といった一部の機器を除き、ほぼ全ての機器類をデジタル制御あるいはモニタするに至った。各機器類を制御するサーバーPCと機器類間のリモートI/Oには必要に応じてYokogawa PLCあるいはNational Instruments FieldPointを用い、またサーバーPCと機器類間の通信インターフェースとしてはEthernet、GPIB、RS232C、(一部計測機器用にUSB)を用いている。サーバーPCとクライアントPC間の通信にはActiveX[1]を用いることで、基本的に任意のクライアントPCからのEPIOS Process Variableを用いたアクセスが可能である(超伝導ウィグラー用磁石電源のように複数PCによるマルチアクセスや他の機器との連携操作の必要のない一部の機器類に対してはシングルPC制御に留めている)。また、近年、一部の加速器構成機器類制御プログラムをシーケンシャルに構築することにより、1台のPCで複数の機器を連動制御させるプログラムの運用も試みている。このような集約的なプログラムは通信障害が発生した場合には特に有効な制御方法である。制御に用いているWindows OSは、PCの購入時期により、Windows 2000、XP、7と上位バージョンのものに順次更新してきた。バージョンの異なるOSが混在した環境であるが、現在までにOSの違いによる制御上の障害は生れていない。しかし、稼働させているサーバープログラムの種類にもよるが、数ヶ月に一度程度システムをリブートする必要があることがある。この問題によりユーザー運転を中断した例は過去にないが、長期的な制御システムの安定性確保はひとつの課題である。Windows OS環境下における制御システムの安定化のためにはPCリブートのルーチン化が簡易で有効ではあるが、原因不明のためメモリーリークの状態などの観測も続けている。

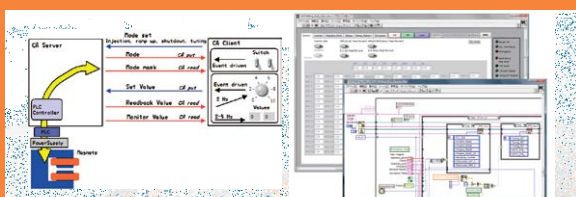
[1] ActiveX CA Tools, <http://www.slaclanford.edu/grp/cd/soft/epios/extensions/ActiveXCA/readme.htm>

System Layout



SAGA-LS加速器制御システムの概略図を示す(Beam LineおよびOffice LANも含む)。近年、Linac制御システムもデジタル制御化されたが図中では省略している。Insertion Deviceは制御室およびBeam Lineサイドからの制御が可能である。Firewallには外部からアクセスができないセキュリティポリシーが設定されている。

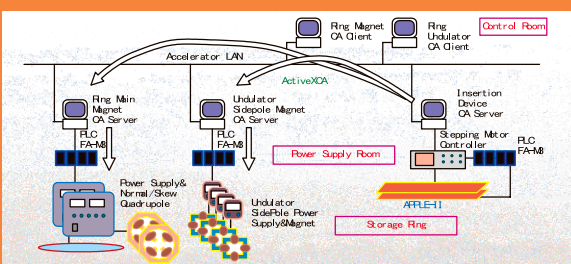
Application Program



SAGA-LS制御システムではMicrosoft WindowsをOSとするコストパフォーマンスに優れたPCにより制御を行っている。通信部には、EPIOSをエミュレートするChannel Access(ActiveX CA)を用いて、サーバークライアント方式により分散された加速器構成機器類を制御している。ActiveX CAは、Windows Vista以降のマシンであっても対応している。アプリケーション開発には主にNational Instruments LabVIEWを用いた。

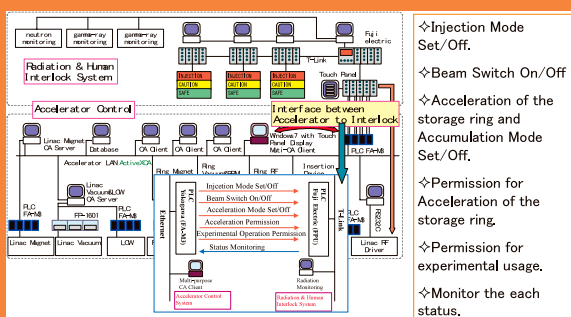
アプリケーションプログラムは、PLCプログラムなど一部はメーカーにより製造されたが、上位制御システムは全てin-houseで実施された。

Integrated Sub-System



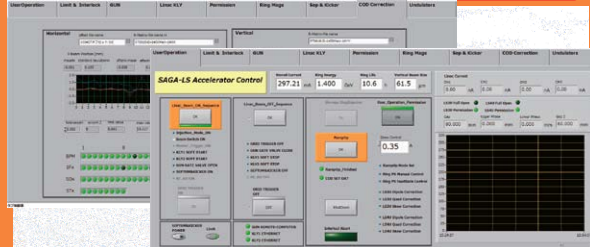
電磁石制御、RF制御、真空モニター等、主要機器の開発の他、外部DOCTによる電磁石電源出力フィードバック制御、4極電磁石磁場中心に対するBPMセンターの校正等も行った。図は、アンジュレータのギャップと位相に連動するフィードフォワードシステムのシステム構成図である。

Interface between accelerator and Interlock



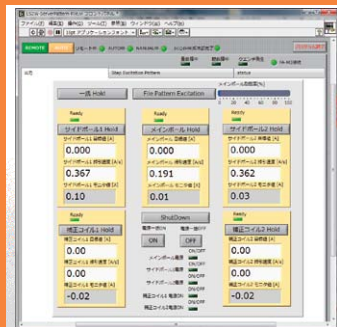
加速器運転の簡易化を目的として両者のインターフェースを設けた。これにより、加速器制御系から、ビームスイッチなどの操作が可能となった。

Multi-purpose CA-Client



加速器制御系のアプリケーションプログラムのいくつかをシーケンシャルに集約し、放射線インターロックシステムと連動したアプリケーションを開発した。Windows7をOSに採用したことにより、タッチディスプレイでの操作が可能である。しかしながら、LabVIEWでは大規模アプリケーションを構築するのは難しい側面があり、現在もクライアントソフトの集約開発を続けている。

超伝導ウィグラー制御



長直線2番(LS2)に設置された超伝導ウィグラー用磁石電源5台の制御を行っている。メインボールにより生成されるBL積を打ち消すように、サイドボールの励磁電流が決められた。サイドボール磁石に巻かれた補正コイルはユーザー運転時に使用していない。励磁の際には、現場PLCのファイルレジスターに収納されたファイルパターンにより通電される。消磁の際は、制御PCより直接的に通電電流が減少される。

他の機器との連携動作の必要性がないため、現在のところ現場制御盤と制御室PCの1対1の制御に留めている。

まとめ

SAGA-LS加速器を構成するほぼ全ての機器類を、PC制御するに至った。サーバーPCとクライアントPCの通信にはActiveXCAを用いている。古いタイプのPCは、Microsoft Windows 2000、新しいタイプのものではWindows 7を用いているが、OSの違いによる障害は生じていない。National Instruments LabVIEWによる開発を行うことで、特に制御のエキスパートではなくても、必要なアプリケーションが短期間で構築できる。LabVIEWの問題点としては大規模なアプリケーションの開発にはあまり適していないこと、過去に作成したプログラムの再利用がそれほど便利ではないことがあげられるかもしれない。また、プログラムの種類にもよるが、メモリーリークを起こしたり、数ヶ月に一度PCをリブートする必要があることが最大のデメリットであろう。メモリーリークは、特にOSのバージョンに関係なく生じている。左図にLabVIEWのメモリーリークの様子をしめす(データベースにアクセスし、表示画面をhtmlで出力させるもの)。加速器の制御PCは基本的に24h稼働させるものなので、安定性確保が課題かと思われる。

